

一、单选

1. 以下说法中正确的是()

- A) C 语言程序总是从第一个的函数开始执行
- B) 在 C 语言程序中, 要调用的函数必须在 `main()` 函数中定义
- C) C 语言程序总是从 `main()` 函数开始执行
- D) C 语言程序中的 `main()` 函数必须放在程序的开始部分

2. 下选项中不是 C 语句的是()。

- A) `{int i; i++; printf( "%d\n", i);}`
- B) `;`
- C) `a=5, c=10`
- D) `{ ; }`

3. 以下选项中不能作为 C 语言合法常量的是

()

- A) `'cd'`
- B) `0.1e+6`
- C) `"\a"`
- D) `'\011'`

4. 若 `x`, `a`, `b` 和 `c` 均是 `int` 型变量, 则执行表达式 `x=(a=1, b=2)` 后 `x` 的结果为()。

- A) 1
- B) 2
- C) 3

D) 不确定

5. 以下能正确定义整型变量 a, b 和 c 并为其赋初值 1 的语句是 ()。

A) `int a=b=c=1;`

B) `int a, b, c=1;`

C) `a=b=c=1;`

D) `int a=1, b=1, c=1;`

6. 字符串“`\t\065\xff\n`”中的字符数(不算‘`\0`’)为()

A) 5

B) 14

C) 8

D) 4

7. 对两个静态函数 A 和 B 进行如下初始化:

```
char A[]="ABCDEF";
```

```
char B[]={'A','B','C','D','E','F'};
```

则下列叙述正确的是()。

A) A 和 B 完全相同

B) A 和 B 只是长度相等

C) A 和 B 不相同, A 是指针数组

D) A 数组长度比 B 数组长

8. 若有以下定义:

```
char a;int b;
```

```
float c;double d;
```

则表达式 $a*b+d-c$ 值的类型为()

- A) float
- B) int
- C) char
- D) double

9. 在 C 语言中，运算对象必须是整型数的运算符是()

- A) %
- B) /
- C) %和\
- D) **

10. 有以下程序

```
main()  
{ int  x, y, z;  
  x=y=1;  
  z=x++, y++, ++y;  
  printf("%d,%d,%d\n", x, y, z);  
}
```

程序运行后的输出结果是

- ()
- A) 2, 3, 3
 - B) 2, 3, 2

C) 2, 3, 1

D) 2, 2, 1

11. 以下选项中，当 x 为大于 1 的奇数时，值为 0 的表达式是()

A) $x\%2==1$

B) $x/2$

C) $x\%2!=0$

D) $x\%2==0$

12. 以下程序的输出结果是()。

```
main()  
{  
    int x=10, y=3;  
    printf("%d\n", y=x/y);  
}
```

A) 0

B) 1

C) 3

D) 不确定的值

13. 若 a 为 `int` 类型，且其值为 3，则执行完表达式 $a+=a-=a*a$ 后， a 的值是()

A) -3

B) 9

C) -12

D) 6

15. 有以下程序

```
main()
{
    char  a='a',b;
    printf("%c,",++a);
    printf("%c\n",b=a++);
}
```

程序运行后的输出结果是

()

A) b,b

B) b,c

C) a,b

D) a,c

16. 设 a、b、c、d、m、n 均为 int 型变量，且 a=5、b=6、c=7、d=8、m=2、n=2，则逻辑表达式 (m=a>b)&&(n=c>d) 运算后，n 的值为()

A) 0

B) 1

C) 2

D) 3

17. 设 x、y、t 均为 int 型变量，则执行语句：x=y=3;t=++x||++y; 后，y 的值为()

A) 不定值

B) 4

C) 3

D) 1

18. 有以下程序段

```
int k=0;
```

```
while(k=1)k++;
```

while 循环执行的次数是()

A) 无限次

B) 有语法错，不能执行

C) 一次也不执行

D) 执行 1 次

19. 能正确表示逻辑关系“ $a \geq 10$ 或 $a \leq 0$ ”的 C 语言表达式是()

A) $a \geq 10$ or $a \leq 0$

B) $a \geq 0 \mid a \leq 10$

C) $a \geq 10 \&\& a \leq 0$

D) $a \geq 10 \mid \mid a \leq 0$

20. 表示关系 $x \leq y \leq z$ 的 c 语言表达式为()

A) $(x \leq y) \&\& (y \leq z)$

B) $(x \leq y) \text{ AND } (y \leq z)$

C) $(x \leq y \leq z)$

D) $(x \leq y) \& (y \leq z)$

21. 设有：`int a=1, b=2, c=3, d=4, m=3, n=3;` 执行
`(m=a>b) || (n=c>d)`后 `n` 的值为()。

- A) 0
- B) 1
- C) 2
- D) 3

22. 以下关于逻辑运算符两侧运算对象的叙述中正确的是
()

- A) 只能是整数 0 或 1
- B) 只能是整数 0 或非 0 整数
- C) 可以是结构体类型的数据
- D) 可以是任意合法的表达式

23. 若 `x` 和 `y` 都是 `int` 型变量，`x=100`，`y=200`，且有下面的程序片段：

```
printf("%d", (x, y));
```

上面程序片段的输出结果是()

- A) 200
- B) 100
- C) 100 200
- D) 输出格式符不够，输出不确定的值

24. 以下程序的输出结果是()

```
main()
```

```
{ int k=17;
```

```
printf("%d, %o, %x \n", k, k, k);  
}
```

- A) 17, 021, 0x11
- B) 17, 17, 17
- C) 17, 0x11, 021
- D) 17, 21, 11

25. x、y、z 被定义为 int 型变量，若从键盘给 x、y、z 输入数据，正确的输入语句是()。

- A) INPUT x、y、z;
- B) scanf("%d%d%d",&x,&y,&z);
- C) scanf("%d%d%d",x,y,z);
- D) read("%d%d%d",&x,&y;&z);

26. 有以下程序

```
main()  
{  
    int i;  
    for(i=0;i<3;i++)  
        switch(i)  
        {  
            case 0: printf("%d",i);  
            case 2: printf("%d",i);  
            default: printf("%d",i);  
        }  
}
```



```
    }  
}
```

程序运行后的输出结果是()

- A) 022111
- B) 021021
- C) 000122
- D) 012

27. C 语言中用于结构化程序设计的三种基本结构是()

- A) 顺序结构、选择结构、循环结构
- B) if、switch、break
- C) for、while、do-while
- D) if、for、continue

29. 在嵌套使用 if 语句时，C 语言规定 else 总是
()

- A) 和之前与其具有相同缩进位置的 if 配对
- B) 和之前与其最近的 if 配对
- C) 和之前与其最近的且不带 else 的 if 配对
- D) 和之前的第一个 if 配对

30. 设有以下程序片段：

```
switch(X)  
{  
    case 'A': printf("A");
```

```
    case 'B': printf("B");  
    default: printf("error");  
}
```

假设 X='A', 程序输出结果是()

- A) A
- B) B
- C) error
- D) AError

31. 读程序:

```
main()  
{ int num=0;  
  while (num<=2)  
  { num++; printf("%d ", num);}  
}
```

上面程序的输出结果是()

- A) 1
- B) 2 2
- C) 1 2 3
- D) 1 2 3 4

33. 以下程序段的执行结果是()。

```
int x=5;  
do {printf("%2d\n", x--);}
```

```
while(!x);
```

- A) 5
- B) 无任何输出
- C) 4
- D) 陷入死循环

34. 设变量已正确定义，则以下能正确计算 $f=n!$ 的程序段是

()

A) $f=0;$

```
for(i=1;i<=n;i++) f*=i;
```

B) $f=1;$

```
for(i=1;i<n;i++) f*=i;
```

C) $f=1;$

```
for(i=n;i>1;i++) f*=i;
```

D) $f=1;$

```
for(i=n;i>=2;i--) f*=i;
```

35. 以下程序的输出结果是()。

```
main()
```

```
{
```

```
    int a, b;
```

```
    for(a=1, b=1; a<=100; a++)
```

```
    {
```

```
        if(b>=10) break;
```

```
        if (b%3==1)
        {
            b+=3;
            continue;
        }
    }

    printf("%d\n", a);
}
```

A) 101

B) 6

C) 5

D) 4

36. 下列合法的数组定义是()。

A) int a[]="string";

B) int a[5]={0, 1, 2, 3, 4, 5};

C) char a="string";

D) char a[]={0, 1, 2, 3, 4, 5};

37. 以下定义语句中，错误的是()

A) int a[]={1, 2};

B) char *a[3];

C) char s[10]="test";

D) int n=5, a[n];

38. 以下正确的数组定义语句是()。

- A) `int y[1][4]={1, 2, 3, 4, 5};`
- B) `float x[3][]={{1}, {2}, {3}};`
- C) `long s[2][3]={{1}, {1, 2}, {1, 2, 3}};`
- D) `int m[1][4]={4};`

39. 以下程序的输出结果是()。

```
main()
{
    int i, a[10];
    for(i=9; i>=0; i--)
        a[i]=10-i;
    printf("%d%d%d", a[2], a[5], a[8]);
}
```

- A) 258
- B) 741
- C) 852
- D) 369

41. 若有定义语句：`int a[3][6];`，按在内存中的存放顺序，a 数组的第 10 个元素是()

- A) `a[0][4]`
- B) `a[1][3]`
- C) `a[0][3]`

D) a[1][4]

42. 以下数组定义中不正确的是()

A) int a[2][3];

B) int b[][3]={0, 1, 2, 3};

C) int c[100][100]={0};

D) int d[3][]={{1, 2}, {1, 2, 3}, {1, 2, 3, 4}};

43. 以下程序的输出结果是()。

```
main()
```

```
{
```

```
    int a[4][4]={ {1, 3, 5}, {2, 4, 6}, {3, 5, 7} };
```

```
    printf("%d%d%d%d\n", a[0][3], a[1][2], a[2][1], a[3][0]);
```

```
}
```

A) 0650

B) 1470

C) 5430

D) 输出值不定

45. 以下程序的输出结果是

()

```
main()
```

```
{    char cf[3][5]={ "AAAA", "BBB", "CC" };
```

```
    printf("\'%s'\n", cf[1]);;
```

```
}
```

- A) "AAAA"
- B) "BBB"
- C) "BBBCC"
- D) "CC"

46. 设有数组定义: `char array [ ]="China"`; 则数组 `array` 所占的空间为()

- A) 4 个字节
- B) 5 个字节
- C) 6 个字节
- D) 7 个字节

47. 以下选项中, 不能正确赋值的是()。

- A) `char s1[10];s1="Ctest";`
- B) `char s2[]={ 'C', 't', 'e', 's', 't' };`
- C) `char s3[20]="Ctest";`
- D) `char *s4="Ctest\n";`

48. 给出以下定义:

`char x[]="abcdnfeg";`

`char y[]={ 'a', 'b', 'c', 'd', 'e', 'f', 'g' };`

则正确的叙述为()。

- A) 数组 `x` 和数组 `y` 等价
- B) 数组 `x` 和数组 `y` 长度相同
- C) 数组 `x` 的长度大于数组 `y` 的长度

D) 数组 x 的长度小于数组 y 的长度

50. 若要求从键盘读入含有空格字符的字符串，应使用函数

()

A) `getc()`

B) `gets()`

C) `getchar()`

D) `scanf()`

51. 有语句：

```
char str1[10],str2[10]={"books"};
```

则能将字符串 books 赋给数组 str1 的正确语句是()。

A) `str1={"Books"};`

B) `strcpy(str1, str2);`

C) `str1=str2;`

D) `strcpy(str2, str1);`

52. 以下语句的输出结果是()

```
printf("%d\n", strlen("\tc\065\xff\n"));
```

A) 5

B) 14

C) 8

D) 输出项不合法，无正常输出

53. 若有语句 `int *point, a=4;` 和 `point=&a;` 下面均代表地址的一组选项是()。

- A) a, point, *&a
- B) &*a, &a, *point
- C) *&point, *point, &a
- D) &a, &*point, point

54. 设已有定义：float x;，则以下对指针变量 p 进行定义且赋初值的语句中正确的是()

- A) float *p = 1024;
- B) int *p = (float)x;
- C) float p = &x;
- D) float *p = &x;

55. 设有语句：int a=1, b=2, *p1=&a, *p2=&b; 以下可使指针 p1 指向变量 b 的赋值语句是()。

- A) p1=*p2
- B) *p1=p2
- C) p1=p2
- D) *p1=*p2

56. 有以下程序

```
main()
{
    int a[]={1, 2, 3, 4, 5, 6, 7, 8, 9, 0}, *p;
    for(p=a; p<a+10; p++)
        printf("%d, ", *p);
```

}

程序运行后的输出结果是()

- A) 1, 2, 3, 4, 5, 6, 7, 8, 9, 0,
- B) 2, 3, 4, 5, 6, 7, 8, 9, 10, 1,
- C) 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
- D) 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

57. 若已定义 `char s[10];` 则在下面表达式中不能表示 `s[1]` 的地址的是()。

- A) `s+1`
- B) `s++`
- C) `&s[0]+1`
- D) `&s[1]`

58. 在以下选项中，操作不合法的一组是()。

- A) `int x[6], *p; p=&x[0];`
- B) `int x[6], *p; *p=x;`
- C) `int x[6], *p; p=x;`
- D) `int x[6], p; p=x[0];`

59. 有以下程序

```
main()
{
    char s[]="159", *p;

    p=s;

    printf("%c", *p++);
}
```

```
    printf("%c", *p++);  
}
```

程序运行后的输出结果是

()

A) 15

B) 16

C) 12

D) 59

61. 以下语句或语句组中，能正确进行字符串赋值的是()

A) char *sp; *sp="right!";

B) char s[10]; s="right!";

C) char s[10]; *s="right!";

D) char *sp="right!";

62. 下面程序段的运行结果是()。

```
char *p= "abcdefgh";
```

```
p+=2;
```

```
printf("%d\n", strlen(strcpy(p, "ABCD")));
```

A) 6

B) 12

C) 4

D) 7

63. 下面程序段的运行结果是()。

```
char str[ ]= "ABC", *p=str;  
printf("%d\n", *(p+3));
```

- A) 67
- B) 0
- C) 字符'C'的地址
- D) 字符'C'

64. 以下程序运行后，输出结果是()

```
main()  
{ char *s="abcde";  
  s+=2;  
  printf("%d\n", s);}
```

- A) cde
- B) 字符c的ASCII码值
- C) 字符c的地址
- D) 出错

66. 有以下程序：

```
main()  
{ char *p[10]={"abc", "aabdfg", "dcdbe", "abbd", "cd"};  
  printf("%d\n", strlen(p[4]));  
}
```

执行后输出结果是()

- A) 2

B) 3

C) 4

D) 5

67. 若有语句: `char *line[5];`, 以下叙述中正确的是()

A) 定义 `line` 是一个数组, 每个数组元素是一个基类型为 `char` 的指针变量

B) 定义 `line` 是一个指针变量, 该变量可以指向一个长度为 5 的字符型数组

C) 定义 `line` 是一个指针数组, 语句中的*号称为间址运算符

D) 定义 `line` 是一个指向字符型函数的指针

68. 设有如下定义: `char *aa[2]={ "abcd", "ABCD" }`; 则以下说法中正确的是()。

A) `aa` 数组成元素的值分别是 "abcd" 和 "ABCD"

B) `aa` 是指针变量, 它指向含有两个数组元素的字符型一维数组

C) `aa` 数组的两个元素分别存放的是含有 4 个字符的一维字符数组的首地址

D) `aa` 数组的两个元素中各自存放了字符串 "abcd" 和 "ABCD" 的首地址

69. 以下函数调用语句中含有()个实参。

`fun((exp1, exp2), (exp3, exp4, exp5));`

A) 1

B) 2

C) 4

D) 5

71. 以下所列的各函数首部中，正确的是()

A) void play(var a :integer, var b:integer)

B) void play(int a,b)

C) void play(int a,int b)

D) Sub play(a as integer,b as integer)

72. C 语言中，函数值类型的定义可以缺省，此时函数值的隐含类型是()

A) void

B) int

C) float

D) double

73. 下面的函数调用语句中 func 函数的实参个数是

()

```
func(f2(v1, v2), (v3, v4, v5), (v6, max(v7, v8)));
```

A) 3

B) 4

C) 5

D) 8

74. 有以下程序

```
fun(int x, int y)
```

```

{ static int  m=0, i=2;

    i+=m+1;    m=i+x+y;    return m;
}

main()

{ int  j=1, m=1, k;

    k=fun(j, m);    printf("%d, ", k);

    k=fun(j, m);    printf("%d\n", k);
}

```

执行后的输出结果是

()

A) 5, 5

B) 5, 11

C) 11, 11

D) 11, 5

75. 以下函数值的类型是

()

```
fun ( float  x )
```

```

{ float  y;

    y= 3*x-4;

    return  y;
}

```

A) int

B) 不确定

C) void

D) float

76. 以下程序的输出结果是

()

```
fun(int x, int y, int z)
{
    z=x*x+y*y;
}

main()
{
    int a=31;

    fun(5, 2, a);

    printf("%d", a);
}
```

A) 0

B) 2

C) 31

D) 无定值

77. 有以下程序

```
void f(int x, int y)
{
    int t;

    if(x<y) { t=x; x=y; y=t; }
}

main()
```



```
{  int  a=4, b=3, c=5;

    f(a, b);    f(a, c);    f(b, c);

    printf("%d, %d, %d\n", a, b, c);

}
```

执行后输出的结果是()

- A) 3, 4, 5
- B) 5, 3, 4
- C) 5, 4, 3
- D) 4, 3, 5

78. 在调用函数时，如果实参是简单变量，它与对应形参之间的数据传递方式是()

- A) 地址传递
- B) 单向值传递
- C) 由实参传给形参，再由形参传回实参
- D) 传递方式由用户指定

79. 有以下程序

```
void f(int b[])

{  int i;

    for (i=2; i<6; i++)

        b[i] *= 2;

}
```

main()

```
{ int a[10]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, i;  
  f(a);  
  for (i=0; i<10; i++)  
      printf("%d, ", a[i]);  
}
```

程序运行后的输出结果是()

- A) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
- B) 1, 2, 6, 8, 10, 12, 7, 8, 9, 10,
- C) 1, 2, 3, 4, 10, 12, 14, 16, 9, 10,
- D) 1, 2, 6, 8, 10, 12, 14, 16, 9, 10,

80. 在 C 语言中，函数返回值的类型最终取决于
()

- A) 函数定义时在函数首部所说明的函数类型
- B) return 语句中表达式值的类型
- C) 调用函数时主调函数所传递的实参类型
- D) 函数定义时形参的类型

81. 调用一个函数，此函数中没有 return 语句，下列说法正确的是：
该函数()。

- A) 没有返回值
- B) 返回若干个系统默认值
- C) 能返回一个用户所希望的函数值
- D) 返回一个不确定的值

83. 若有以下调用语句, 则不正确的 fun 函数的首部是()

```
main()
{
    ...
    int a[50], n;
    ...
    fun(n, &a[9]);
    ...
}
```

- A) void fun(int m, int x[])
- B) void fun(int s, int h[41])
- C) void fun(int p, int *s)
- D) void fun(int n, int A)

85. 有以下程序

```
void swap(char *x, char *y)
{
    char t;
    t=*x; *x=*y; *y=t;
}

main()
{
    char *s1="abc", *s2="123";
    swap(s1, s2); printf("%s,%s\n", s1, s2);
}
```

程序执行后的输出结果是

()

A) 123, abc

B) abc, 123

C) 1bc, a23

D) 321, cba

86. 以下程序的输出结果()

```
#include<stdio.h>
```

```
sub(int x,int y,int * z)
```

```
{
```

```
  *z=y-x;
```

```
}
```

```
main()
```

```
{
```

```
  int a, b, c;
```

```
  sub(10,5,&a);
```

```
  sub(7,a,&b);
```

```
  sub(a,b,&c);
```

```
  printf("%d,%d,%d\n", a,b,c);
```

```
}
```

A) 5, 2, 3

B) -5, -12, -7

C) -5, -12, -17

D) 5, -2, -7

87. 已定义以下函数

```
int fun( int *p)
{ return *p; }
```

fun 函数返回值是

()

A) 不确定的值

B) 一个整数

C) 形参 p 中存放的值

D) 形参 p 的地址值

88. 已定义以下函数：

```
fun (char* p2, char* p1)
{ while ((*p2=*p1) != '\0') {p1++;p2++;} }
```

函数的功能是

()

A) 将 p1 所指字符串复制到 p2 所指内存空间

B) 将 p1 所指字符串的地址赋给指针 p2

C) 对 p1 和 p2 两个指针所指字符串进行比较

D) 检查 p1 和 p2 两个指针所指字符串中是否有 '\0'

89. 有以下程序

```
#include <stdio.h>
```

```
int fun(int a, int b)
```

```
{ if (b==0)    return a;

    else      return (fun(--a, --b));

}
```

```
main()
```

```
{ printf("%d\n", fun(4, 2)); }
```

程序的运行结果是()

- A) 1
- B) 2
- C) 3
- D) 4

90. 在函数调用过程中，如果函数 funA 调用了函数 funB，函数 funB 又调用了函数 funA，则()

- A) 称为函数的直接递归调用
- B) 称为函数的间接递归调用
- C) 称为函数的循环调用
- D) C 语言中不允许这样的递归调用

91. 以下叙述中正确的是()

- A) 全局变量的作用域一定比局部变量的作用域范围大
- B) 静态(static)类别变量的生存期贯穿于整个程序的运行期间
- C) 函数的形参都属于全局变量
- D) 未在定义语句中赋初值的 auto 变量和 static 变量的初值都是随机值

92. 以下程序的输出结果是

()

```
int  a,b;

void fun()

{   a=100;  b=200;   }

main()

{   int  a=5, b=7;

    fun();

    printf("%d%d \n", a,b);

}
```

A) 100200

B) 57

C) 200100

D) 75

93. 以下程序的输出结果是

()

```
int f()

{   static int i=0;

    int  s=1;

    s+=i;  i++;

    return  s;

}
```

```
main()
{   int   i, a=0;
    for(i=0; i<5; i++) a+=f();
    printf("%d\n", a);
}
```

- A) 20
- B) 24
- C) 25
- D) 15

二、多选

1. 下列定义变量的语句中正确的是()

- A) `int _int;`
- B) `double int_;`
- C) `char For;`
- D) `float US$;`

2. 设有定义：`int k=1, m=2; float f=7;`，则以下选项中符合 C 语言语法的表达式是

()

- A) `k=k>=k`
- B) `-k++`
- C) `k%int(f)`
- D) `k<>m`

3. 若有定义语句: `double x[5]={1.0, 2.0, 3.0, 4.0, 5.0}`, `*p=x`; 则正确引用 `x` 数组元素的是()

A) `*p`

B) `x[5]`

C) `*(p+1)`

D) `*x`

4. 若要求定义具有 10 个 `int` 型元素的一维数组 `a`, 则以下定义语句中正确的是

()

A) `#define N 10`

`int a[N];`

B) `#define n 5`

`int a [2*n];`

C) `int a[5+5];`

D) `int n=10, a[n];`

5. 按照 C 语言规定的用户标识符命名规则, 能出现在标识符中的是

()

A) 大写字母

B) 连接符

C) 数字字符

D) 下划线

6. 以下不合法的字符型常量是()

A) ' \x13'

B) ' \081'

C) ' \065'

D) "\n"

7. 以下合法的字符常量是

()

A) ' \018'

B) ' \'"'

C) ' \\'

D) ' \0xcc'

8. 以下叙述中正确的是()

A) C 语句必须以分号结束

B) 复合语句在语法上被看作一条语句

C) 空语句出现在任何位置都不会影响程序运行

D) 赋值表达式末尾加分号就构成赋值语句

9. 以下合法的赋值语句是()

A) n=(i=2, ++i);

B) j++;

C) ++(i+1);

D) x=j>0;

10. 以下能正确定义二维数组的选项是

()

A) `int a[2][2] = {{1}, {2}};`

B) `int a[ ][2] = {1, 2, 3, 4};`

C) `int a[2][2] = {{1}, {2}, {3}};`

D) `int a[2][ ] = {{1, 2}, {3, 4}};`

11. 若有定义：`int aa[8];` 则以下表达式中能代表数组元 `aa[1]` 的地址的是

()

A) `&aa[0]+1`

B) `&aa[1]`

C) `&aa[0]++`

D) `aa+1`

12. 已有定义：`int i, a[10], *p;` , 则不合法的赋值语句是()

A) `p=100;`

B) `p=a[5]`

C) `p=&a[2]+2`

D) `p=a+2;`

13. 以下能正确进行字符串赋初值的语句是()

A) `char str[5]="good!";`

B) `char str[]="good!";`

C) `char *str="good!"`;

D) `char str[5]={ 'g', 'o', 'o', 'd' }`;

14. 已知大写字母 A 的 ASCII 码是 65，小写字母 a 的 ASCII 码是 97。

以下能将变量 c 中的大写字母转换为对应小写字母的语句是()

A) `c=(c-'A')%26+'a'`

B) `c=c+32`

C) `c=c-'A'+'a'`

D) `c=('A'+c)%26-'a'`

15. 设有以下定义和语句

```
char str[20]="Program", *p;
```

```
p=str;
```

则以下叙述中错误的是()

A) *p 与 str[0] 中的值相等

B) 可以执行 `p++` 操作

C) 可以执行 `str++` 操作

D) str 数组长度和 p 所指向的字符串长度相等

16. 对于下面①，②两个循环语句，正确的描述是()。

① `while(1);`

② `for( ; ; );`

A) ①是无限循环

B) ②是无限循环

C) ①循环一次

D) ②循环一次

17. 以下错误的函数定义形式是()

A) `double fun (int x , int y)`

B) `double fun (int x ; int y)`

C) `double fun (int x , int y) ;`

D) `double fun (int x , y) ;`

18. 对于基本类型相同的两个指针变量之间，可以进行的运算是

()

A) `<`

B) `=`

C) `+`

D) `-`

19. 能把字符串:Hello!赋给数组 b 的语句是()

A) `char b[10]={ 'H', 'e', 'l', 'l', 'o', '!', ' ' };`

B) `char b[10];b="Hello!";`

C) `char b[10];strcpy(b, "Hello!");`

D) `char b[10]="Hello!";`

20. 以下能对二维数组 a 进行正确初始化的语句是()

A) `int a[2][3]={0};`

B) `int a[2][ ]={{1, 2}, {0}};`

C) `int a[2][3]={ {1, 2}, {3, 4}, {5, 6} };`

D) `int a[ ][3]={1, 2, 3, 4, 5, 6};`

21. `sizeof(double)` 是

A) 一种函数调用

B) 一个双精度型表达式

C) 一个整型表达式

D) `sizeof` 是运算符

22. 设 `a` 为整型变量，能正确表达数学关系： $10 < a < 15$ 的 C 语言表达式是()

A) `10 < a < 15`

B) `a==11 || a==12 || a==13 || a==14`

C) `a>10&&a<15`

D) `(a<10) || (a>15)`

三、判断

1. C 程序的基本组成单位是函数

2. 每个 C 程序中都必须要有个 `main()` 函数

3. C 程序中注释部分可以出现在程序中任意合适的地方

4. C 程序的执行总是从 `main` 函数开始，在 `main` 函数结束

5. `++(i+1);` 是非法的赋值语句

6. C 语言中有逻辑类型

7. 可以用关系运算符对字符串的大小进行比较

8. 设有定义语句：`char b= '\123'`; 则变量 `b` 包括 4 个字符

9. 若有定义语句：char s[10]="1234567\0\0";，则 strlen(s)的值是 9
10. a=b+c=1 是正确的赋值表达式
11. 对于单目运算符++、--，它们的运算对象可以是任何变量和常量
12. 表达式：10!=9 的值是 true
13. 表达式 1||2||3||4 的值是 1。
14. sizeof(float)的值是 4
15. if(x<y) {x++;y++;} 是正确的 if 语句。
16. 用 do-while 语句构成的循环, 在 while 后的表达式为零时结束循环
17. 对 for(表达式 1; ; 表达式 3)可理解为 for(表达式 1; 0; 表达式 3) 。
18. break 语句只能用于 switch 语句体中
19. 当程序执行中，数组元素的下标超出所定义的下标范围时，系统将给出“下标越界”的出错信息
20. 假定 int 类型变量占用两个字节，其有定义：int x[10]={0, 2, 4};，则数组 x 在内存中所占字节数是 6
21. 若有定义：int a[2][3];对 a 数组元素正确引用的是 a[2][3]
22. 若有说明 int s[3][4]={0};则只有元素 s[0][0] 可得到初值 0。
23. 不能在赋值语句中通过赋值运算符“=”对字符型数组进行整体赋值
24. 函数中的形式参数是局部变量

25. 函数的定义和函数的调用均可以嵌套